Using Intermediate States to Improve the Ability of the Arden Syntax to Implement Care Plans and Reuse Knowledge

Eric H. Sherman, M.D., M.S., George Hripcsak, M.D., Justin Starren, M.D., Robert A. Jenders, M.D., M.S., Paul Clayton, Ph.D.

Department of Medical Informatics
Columbia University, New York, NY

ABSTRACT

The Arden Syntax is one of a few knowledge representation languages currently in use for clinical decision support. While some of these languages are being used in active patient care settings, none have gained widespread acceptance as a clinical tool. Prior attempts to represent temporally complex care plans in the Arden Syntax have revealed difficulties in representing and tracking series of consecutive time-oriented events and recommendations, in sharing and reusing knowledge and in dealing with unobtainable data. In an attempt to improve Arden's ability to deal with these problems and demonstrate the importance of these factors, the clinical event monitor has been adapted to store coded data representing Intermediate States in the Columbia Presbyterian Medical Center (CPMC) central data repository. The Intermediate States define the current state of the patient as laid out in the care plan. Four care plans were constructed. The findings include an improved ability to track complex series of events and recommendations over long periods of time. The knowledge generated by the electronic care plans was able to be reused by the care plan that generated it, by other elements of the knowledge base and by non-decision support applications. Modular development, facilitated by the changes, simplified dealing with data not available to the central data repository by aiding the implementation of those parts of the care plan for which sufficient data is available.

INTRODUCTION

Clinical practice guidelines and care plans are being produced and published with increasing frequency. Justification for their popularity has been demonstrated in studies that document improvement in outcomes [1] when they are used to guide patient care. The inability of users to remember guidelines and maintain the improvement in patient care has led to efforts to reproduce and maintain the improvement in outcomes through the automation of guidelines and care plans. There are already numerous examples of guidelines and care plans that have been automated [2].

In contrast, there has been only slow movement towards placement of automated guidelines and care plans (which will be referred to collectively as care plans) into everyday medical practice where they can most directly affect care. This is particularly true of the more temporally complex care plans where the patients must be followed over the life of a disease or treatment plan. While some of the reasons for this are external to the knowledge base and its environment (such as the paucity of positive cost-effectiveness studies), improvements in the ability to create and share knowledge bases will surely help prove the usefulness and spread the use of automated decision support.

Care plans create special difficulties for knowledge engineers and the knowledge representations they use. Knowledge representation formalisms have been compared with respect to their ability to address these areas of functional complexity [3,4]. Some of these knowledge representation formalisms have been specifically designed to reduce one or more type of complexity (e.g. Arden Syntax goals included the transfer of rules between institutions and a simplified coding scheme [5]; some declarative languages have a goal of reuse of knowledge). An unfortunate side effect has usually been to increase another type of complexity (e.g. Arden has difficulty reusing logic; declarative languages have less intuitive coded schemes).

Based upon our prior experience with the Arden Syntax and comparison studies of knowledge representation formalisms [3,4,6], we have identified a core set of functional requirements for knowledge representation tools that have particular importance when automating temporally complex knowledge bases. These include temporal interaction, ease of encoding, sharing and reuse of knowledge, and dealing with unobtainable data. We hypothesize that by changing the paradigm for the way knowledge derived from Arden Medical Logic Modules (MLMs) is handled (i.e. by storing it as coded data in the central data repository) we can demonstrate an improvement in the way Arden Syntax care plans are developed and used. We have built computerized care plans in four areas to evaluate the benefits of the new capabilities.

0195–4210/95/$5.00 © 1995 AMIA, Inc.

FUNCTIONAL REQUIREMENTS IN ARDEN

Temporal Interaction
To successfully automate care plans a knowledge representation must not only be able to encode the knowledge in the care plan but it must also be able to deal with events that occur over time. This results in an ever changing patient state representation. The electronic version of a care plan must be able to know when to "wake up" to reevaluate the patient state, how to handle unpredictable events and how to maintain a history of provider/care plan interactions.

Being a procedural language Arden has little difficulty representing complex temporal events. It does not need to cover all possible occurrences so as not to get lost in the logic of the care plan (a problem that has led some to approach knowledge acquisition with exhaustive decision tables to assure completeness). But Arden, having been designed to divide knowledge into rules representing individual decisions, is not a traditional rule-based production system since it does not have an explicit working set of knowledge. Therefore, in theory, it is not designed to handle series of related decisions. While tools already exist in the CPMC Arden Syntax compiler to call other MLMs and store text in the database, the goal is to enhance the ability of Arden MLMs to interact and maintain a knowledge of the patient's state, i.e. her place in the care plan.

Ease of Use
Arden is one of a group of procedural languages designed for representing medical knowledge. Among the design goals that prompted the choice of a procedural language and the "single rule, single decision" paradigm was the importance of creating a knowledge acquisition language that was easy to use. While the developers never claimed that the syntax would be easy enough for novice users to program, a specified goal was that the average health care provider be able to read the code and understand its use. Arden becomes more difficult to use when attempting to tie together multiple rules that might fire in unpredictable patterns. Any improvement in the storage and retrieval of derived knowledge must be evaluated with an eye to discovering any loss in the ease with which the code can be written and understood.

Knowledge sharing and reuse
The transfer of knowledge across institutions was of particular interest to the designers of Arden. Unfortunately, the ability of MLMs to be transferred to a new setting is limited [7]. Shwe et al [4] describe the difficulty of reusing knowledge in the Arden Syntax due to its procedural representation. They note that MLMs may be shareable (e.g. between institutions) but that the unit of reuse of the knowledge is the MLM itself. They demonstrated how medical knowledge within the MLM cannot be reused elsewhere and they describe how a declarative knowledge representation is better able to allow the reuse of the knowledge base.

Different automated care plan often utilize bits of derived knowledge, such as cardiac risk scores or diagnoses. This requires either that the knowledge from one section be available to other sections (and other care plans) or that the knowledge be rederived so that the state of the patient can be understood. Clearly, the latter solution is inefficient and unacceptable. Arden needs to be able to efficiently reuse knowledge derived in a care plan before it is usable as a knowledge representation for this purpose.

Unavailable Data
It is unlikely that any time soon a computer information system will claim to have a complete set of patient information in a coded format. Among the well known obstacles are encoding free text notes and getting health care providers to enter data. When evaluating care plans for possible translation into automated form a knowledge engineer must determine if the data necessary is being collected. If not, the care plan is often considered unusable. For example, a patient with an elevated cholesterol should be placed on diet therapy prior to considering drug therapy. At CPMC, no data regarding diet therapy is currently obtained. Therefore, it is impossible to determine when to suggest drug therapy.

Translating a care plan into the Arden Syntax with its atomization of decisions, has provided the possibility of coding only those parts of the care plan for which the data and intermediate conclusions needed from other elements of the care plan exist. But remaining problems include how to connect the fragmented elements (the MLMs) and, once connected, how to add new MLMs without having to reengineer the entire care plan.

A NEW USE OF ARDEN AND ITS POTENTIAL EFFECT

The Arden Syntax was originally designed to be used as a set of independent modules. The paradigm was one module that addressed one decision that, if satisfied, resulted in an action. A limited ability to allow interaction between rules was permitted. The functions associated with this include MLM nesting (synchronous and asynchronous call statements that evoke MLMs), and storage of text strings in the database. Experimentation with practice guidelines and care plans using these functions has been successfully carried out (the knowledge could be represented), but the complexity and limitations of this approach have been evident.

The new approach changes how data is stored in the database. A new interface between the knowledge base and the database permits the storage of a coded attribute associated with a value. Each coded attribute must be a concept in the local medical vocabulary - at CPMC this is the Medical Entities Dictionary (MED). The value can be any legal Arden Syntax data type or a local code. All data stored in the database are associated with a primary time that indicates when the MLM stored the data. A typical use is the storage of a code that represent the start of a protocol with a boolean value of true or false that indicates whether the patient is being placed on the protocol or being removed from the protocol. Another typical example is scheduling. Each event to be scheduled is stored as its representative MED code and a time value indicating when the event is expected to occur. Also, codes can be used to indicate the current state of the patient, her place within a care plan or guideline, by storing the MED code whose concept is that state.

Using this new capability, temporal complexity should be more easily dealt with. The storage of coded Intermediate States in the database provides the care plan MLMs with a working set similar to rule-based production and blackboard [8] systems. Therefore, events that change a patient's place in the care plan should be tracked easily and the next decision made without requiring a complex network of MLMs that directly call other MLMs.

While combining separate MLMs into a related network in the Arden Syntax will certainly be more difficult than building unrelated MLMs, we can expect an improvement in the ease of use over our current methods of building care plans if each rule can be developed, debugged and run independently. Since each MLM represents a separate decision within the care plan, with its own particular evoking trigger (the piece of data that caused an MLM to be evaluated), as long as the trigger can be obtained from the database, that piece of the care plan can be produced, tested, run and evaluated as an independent module. Modular development means that a care plan could be produced piecemeal and that the order in which these pieces are developed need not be predetermined. This frees the knowledge engineer from the need to organize the entire electronic care plan prior to attacking any of the parts.

Knowledge reuse should be greatly facilitated by these methods, but knowledge sharing (or transfer) is not likely to be enhanced. The Intermediate States are expected to be a significant source of reuse of medical knowledge. While the code itself is not likely to be any more reusable than it would be prior to these changes, the live interaction of the code and data that produces patient specific knowledge should be amenable to multiple types of reuse. First, other MLMs within a care plan must be able to make use of the stored Intermediate States in order for care plans to be successfully represented in this scheme. Second, other MLMs in the knowledge base could use these Intermediate States as starting data or as triggers to evoke their logic. Third, these Intermediate States could be used by applications other than decision support, because they are stored in standardized form in our central data repository.

Finally, an enhanced ability to deal with unobtainable data would demonstrate another advantage to Arden of this new approach. The ability to build care plans in a modular fashion, as described above, is expected to allow the engineer to produce and activate a piece of a care plan even if a part of the care plan that precedes it is missing. While this would not be expected to be feasible in all occasions, since direct dependencies occur between some decisions in a care plan, there should be examples where these dependencies do not exist.

SAMPLE CARE PLANS - AN ANALYSIS

The first plan, for inpatient use of Warfarin therapy is diagrammed in Figure 1. The Warfarin care plan's construction demonstrates how a care plan can be built and activated piecemeal. The effort was only minimally greater than that of single MLMs due the simplicity and uniformity of the additional database calls used to add and query the Intermediate States. As is clear in Figure 1, the MLMs have no direct connections between them. But there are interdependencies. MLM #2 is completely dependent on MLM #1, since MLM #2 is only evoked when MLM #1 determines that the Intermediate State "New Warfarin Rx" exists. But MLM #3 was built independently since it does not depend on the other MLMs to activate it, although its conclusions do update the state of the patient. MLM #4 while temporally parallel to MLM #2 and prior to MLM #3 does not depend on any of them to be triggered and does not change the state of the patient in a way that caused MLMs #2 and #3 to be dependant upon it for testing or implementation. Therefore, MLM #4 can be built and added to the Warfarin care plan at any time.

In addition, the Warfarin care plan demonstrates how Intermediate States simplify the temporal complexity

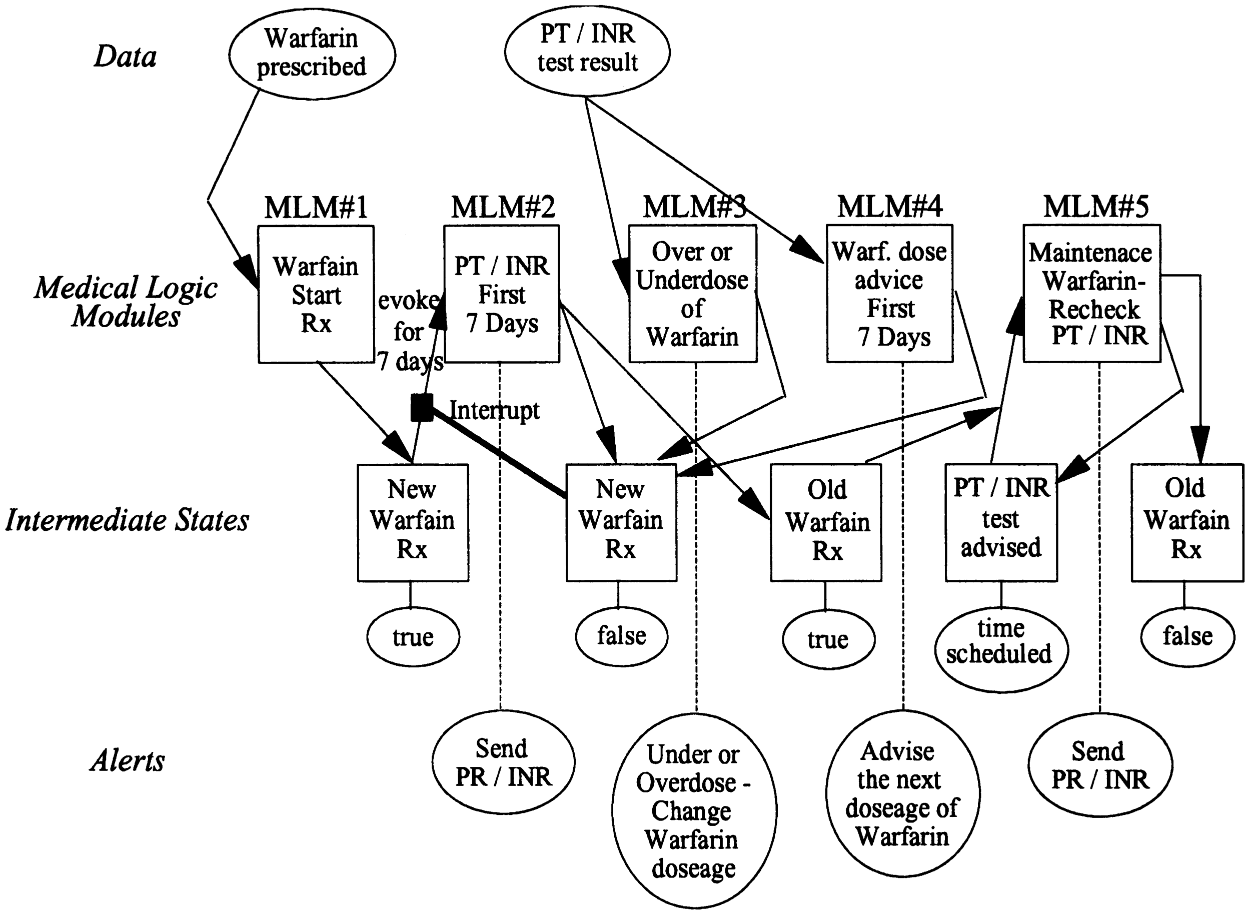

Figure 1: The Warfarin Care Plan: All arrows entering an MLM from the left side are evoking the MLM. All arrows leaving the MLM are storing Intermediate States in the Central Data Repository.

of the care plan. Although a patient started on Warfarin therapy is designated as on protocol for "New Warfarin Rx," as soon as the value of this designation is changed to false (off protocol) it needs to feedback on MLM #2 to stop it from executing. Not only can this change in patient status come from multiple MLMs (#2, #3, #4), but it can also come at any time. By simply changing the value of one Intermediate State rather then networking the assorted MLMs, the development process was significantly simplified.

The Cholesterol care plan implementation demonstrates the ability to develop parts of a care plan despite unobtainable data. The Cholesterol care plan has critical Intermediate States representing the assorted therapy options such as Diet I Rx suggested, Diet II begun, Drug Rx suggested and Drug Rx begun. But the status of diet therapy is currently unknowable, since no health care providers are providing the database with this information. Despite diet therapy being involved in a series of critical middle steps in the cholesterol care plan, an analysis revealed that while the suggestion to move to drug therapy is dependent on knowing if the patient is on diet therapy, the management of drug therapy is not dependent on the knowledge of previous diet therapies. Therefore, the early parts of the care plan have been built and the late parts of the care plan been built, while the middle must wait until data on diet therapy is available before it can be activated and tested. Since only Intermediate States are used to transfer the knowledge derived from one MLM to another, no network of inter-MLM calls need to be a part of the completed care plan. Therefore, the other MLMs that will be part of the completed care plan can be built and activated at a later date, without requiring any adaptation to the MLMs already running.

The reuse of knowledge stored as Intermediate States in the central data repository was demonstrated in two groups of MLMs, one for determination of the etiology of acute renal failure and a health maintenance screening module. The former involved the addition of new MLMs to one of the original CPMC MLMs, "Creatinine Monitor," that diagnoses acute renal failure (ARF). Each of the new MLMs tests for one of the groups of iatrogenic etiologies of ARF (e.g. medications and radiologic dye studies). The original MLM was modified to store its result in coded form. This code is used as a trigger to investigate its cause.

Another example of reuse of knowledge derived from the knowledge base comes from the MLM "Health Maintenance List." This MLM is the starting point of a series of MLMs that encourage compliance with health maintenance screening. The "Health Maintenance List" MLM determines which health maintenance screening tests are appropriate for an individual (e.g. mammography is only appropriate in females within an age range) and stores the coded list in the database. Other MLMs, one for each health maintenance screening test, alert the health care provider if the screening test is not accomplished in a timely fashion. These latter MLMs are triggered initially when the MED code corresponding to that health maintenance screening test is stored. Many of the MLMs for the individual screening tests have be designed to fire periodically (e.g. mammograms are suggested every 1-2 years). The storage of a Health Maintenance list, this time without the MED code for a particular health maintenance screening test "turns off" the appropriate MLM when it detects the missing code in the most recent list in the database.

Intermediate States can be used by applications other than decision support. A workstation for health care providers in the outpatient setting has been built that queries the central data repository for a list of the clinical appropriate health maintenance screening tests derived from the Health Maintenance care plan. This knowledge is routinely displayed in conjunction with the date each activity was last performed.

DISCUSSION

The technique used to adapt the use of the Arden Syntax to a particular need of the knowledge engineers at CPMC is particularly instructive. First, attempts were made to use the existing application to satisfy the new requirements. The identification of a key set of limitations permitted us to identify the source of our problem, the difficulty of interaction between MLMs, and to draw up potential solutions. Implementation of one of these solutions was followed by a study to determine if these key limitations were addressed. The difficulty (and the danger) of adapting a knowledge representation is to not negatively effect the benefits inherent in that formalism.

The results of this adaptation has been to simplify the use of the Arden Syntax (without changing the syntax itself) for the purpose of represented care plans and practice guidelines, and to extend the usefulness of care plans and practice guidelines by allowing the knowledge derived about patients to be used systemwide.

The usefulness of logic composed in the Arden Syntax has been expanded by permitting the reuse of the knowledge created. This work has demonstrated three ways in which this knowledge can be shared. The first two, sharing of knowledge of the patient among the MLMs of a care plan and unrelated MLMs in the knowledge base, expand the uses of Arden towards that of other more complex knowledge representations. Perhaps more important, are uses of the knowledge outside the decision support application. The example presented here showed the creation, by the knowledge base, of lists of tasks that a workstation application uses to keep its users informed. Another application would present those Intermediate States that correspond to diagnoses to the primary care giver for confirmation. The information system Problem List (or analogously, medication and allergy lists) could be automatically populated from an assortment of data sources (e.g. pathology reports, discharge summaries). What other future uses this new capability will provide will reveal themselves in time.

BIBLIOGRAPHY

1. Grimshaw JM, Russel IT. Effect of Clinical Guidelines on Medical Practice: a Systematic Review of Rigorous Evaluations. Lancet 1993; 342:1317-22.
2. Johnston ME, Langton KB. Haynes RB, Mathieu A. Effects of Computer-based Clinical Decision Support Systems on Clinician Performance and Patient Outcome. Ann Intern Med 1994; 120:135-142.
3. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Computers and Biomedical Research 1994; 27:291-324.
4. Shwe M, Sujansky W, Middleton B. Reuse of Knowledge Represented in the Arden Syntax. In: Frisse M ed. Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care. New York: McGraw-Hill, 1992: 47-51.
5. Starren J, Xie G. Comparison of Three Knowledge Representation Formalisms for Encoding the NCEP Cholesterol Guidelines. In: Ozbolt JG ed. Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care. New York: McGraw-Hill, 1994: 792-796.
6. Van der Lei J, Musen MA. The Separation of Reviewing Knowledge form Medical Knowledge. Meth Inform Med 1995; 34: 131-139.
7. Pryor TA, Hripcsak G. Sharing MLM's: An Experiment between Columbia-Presbyterian and LDS Hospital. In Safran C ed. Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care. New York: McGraw-Hill, 1993: 399-403.